

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Atsuhiko OTAKA et al.**

Serial Number: **Not Yet Assigned**

Filed: **December 16, 2003**

Customer No.: 38834

For: **REDUNDANCY MANAGEMENT METHOD FOR BIOS, DATA PROCESSING
APPARATUS AND STORAGE SYSTEM FOR USING SAME**

CLAIM FOR PRIORITY UNDER 35 U.S.C. 119

Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

December 16, 2003

Sir:

The benefit of the filing date of the following prior foreign application is hereby requested for the above-identified application, and the priority provided in 35 U.S.C. 119 is hereby claimed:


Japanese Appln. No. 2002-365618, filed on December 17, 2002

In support of this claim, the requisite certified copy of said original foreign application is filed herewith.

It is requested that the file of this application be marked to indicate that the applicants have complied with the requirements of 35 U.S.C. 119 and that the Patent and Trademark Office kindly acknowledge receipt of said certified copy.

In the event that any fees are due in connection with this paper, please charge our Deposit Account No. 50-2866.

Respectfully submitted,
WESTERMAN, HATTORI, DANIELS & ADRIAN, LLP


William F. Westerman
Reg. No. 29,988

Atty. Docket No.: 032172
Suite 700
1250 Connecticut Avenue, N.W.
Washington, D.C. 20036
Tel: (202) 822-1100
Fax: (202) 822-1111
WFW/yap

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2002年12月17日
Date of Application:

出願番号 特願2002-365618
Application Number:

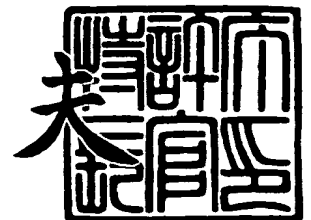
[ST. 10/C]: [JP 2002-365618]

出願人 富士通株式会社
Applicant(s): 株式会社PFU

2003年10月24日

特許庁長官
Commissioner,
Japan Patent Office

今井 康夫



【書類名】 特許願

【整理番号】 0252646

【提出日】 平成14年12月17日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 13/10 330

【発明の名称】 B I O S の冗長管理方法、データ処理装置及びストレージシステム

【請求項の数】 5

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 大高 敦弘

【発明者】

 【住所又は居所】 石川県河北郡宇ノ気町字宇野気ヌ98番地の2 株式会社ピーエフユー内

 【氏名】 宮下 勇

【発明者】

 【住所又は居所】 石川県河北郡宇ノ気町字宇野気ヌ98番地の2 株式会社ピーエフユー内

 【氏名】 田中 法美

【特許出願人】

 【識別番号】 000005223

 【氏名又は名称】 富士通株式会社

【特許出願人】

 【識別番号】 000136136

 【氏名又は名称】 株式会社ピーエフユー

【代理人】

【識別番号】 100094514

【弁理士】

【氏名又は名称】 林 恒徳

【選任した代理人】

【識別番号】 100094525

【弁理士】

【氏名又は名称】 土井 健二

【手数料の表示】

【予納台帳番号】 030708

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9704944

【包括委任状番号】 0215696

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 B I O S の冗長管理方法、データ処理装置及びストレージシステム

【特許請求の範囲】

【請求項 1】 O S がハードウェアを使用できる環境に、前記ハードウェアを設定する B I O S を、各々格納する一対のメモリの一方を稼動中、他方を待機に使用するステップと、

前記一方のメモリの B I O S がブート不能である時に、前記待機中のメモリの B I O S に切り換えるステップと、

前記 B I O S のアップデートを前記待機中のメモリへの書込みにより実行するステップとを有する

ことを特徴とする B I O S の冗長管理方法。

【請求項 2】 前記待機中のメモリへの前記 B I O S のアップデートが成功した時に、前記待機中のメモリを前記稼動中に切り換えることを許可するステップを更に有する

ことを特徴とする請求項 1 の B I O S の冗長管理方法。

【請求項 3】 前記切り換え後、前記稼動中に切り換えられたメモリの B I O S を、前記待機中に切り換えられたメモリに、書込み、冗長化するステップを更に有する

ことを特徴とする請求項 2 の B I O S の冗長管理方法。

【請求項 4】 C P U を含むハードウェアと、

O S が前記ハードウェアを使用できる環境に、前記ハードウェアを設定する B I O S を、各々格納する一対のメモリと、

前記ハードウェアの起動時に、前記一対のメモリの一方を稼動中、他方を待機に使用し、前記一方のメモリの B I O S がブート不能である時に、前記待機中のメモリの B I O S に切り換えるサービスプロセッサとを有し、

前記 C P U は、前記 B I O S のアップデートを前記待機中のメモリへの書込みにより実行する

ことを特徴とするデータ処理装置。

【請求項 5】 C P U を含むハードウェアと、 O S が前記ハードウェアを使用できる環境に、前記ハードウェアを設定する B I O S を、各々格納する一対のメモリと、前記ハードウェアの起動時に、前記一対のメモリの一方を稼動中、他方を待機に使用し、前記一方のメモリの B I O S がブート不能である時に、前記待機中のメモリの B I O S に切り換えるサービスプロセッサとを有するストレージ制御装置と、

前記ストレージ制御装置に接続された複数のストレージ装置とを有し、

前記ストレージ制御装置の前記 C P U は、前記 B I O S のアップデートを前記待機中のメモリへの書込みにより実行する

ことを特徴とするストレージシステム。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、 B I O S (Basic Input/Output System) を冗長管理する B I O S の冗長管理方法、データ処理装置、及びストレージシステムに関し、特に、 B I O S メモリを 2 つ実装した B I O S の冗長管理方法、データ処理装置及びストレージシステムに関する。

【 0 0 0 2 】

【従来の技術】

磁気ディスク、光磁気ディスク、光ディスク等の記憶媒体を利用したストレージ機器では、データ処理装置の要求で、記憶媒体を実アクセスする。データ処理装置が、大容量のデータを使用する場合には、複数のストレージ機器と制御装置とを備えたストレージ装置を利用する。

【 0 0 0 3 】

このようなストレージ装置では、保存データの信頼性や、装置の信頼性を向上するため、冗長構成を採用している。更に、ストレージ制御装置は、 C P U を含むデータ処理装置で構成されている。 C P U は、 C P U の制御プログラムである O S (Operating System) によって、資源の割り付けと保護、プログラムの実行、入出力操作、ファイル操作等の様々なサービスをユーザーに提供する。

【 0 0 0 4 】

これらのサービスを実現するためのOSの基本部分は、カーネル(kernel)と呼ばれている。近年のOS, 特にパーソナル・コンピュータ用OSは、異なるハードウェアでも、共通のOSが動作するように、ハードウェアを制御する部分と、そうでない部分とを別モジュール群で作成している。このハードウェア制御部分は、BIOS (Basic Input/Output System) と称され、そうでない部分をカーネルと称される。

【 0 0 0 5 】

BIOSは、コンピュータシステムのハードウェアをチェックし、カーネルがハードウェアを使用できる環境設定を行う。このようなBIOSは、従来、ファームウェアにバインドされ、メモリに一括格納されるため、版数が異なるものが動作することを防止していた(例えば、特許文献1参照)。

【 0 0 0 6 】**【特許文献1】**

特開平11-306007号公報

【 0 0 0 7 】**【発明が解決しようとする課題】**

一方、ある型番のCPUでも、バグ等の発生により改良したCPUも同一の型番を付す、いわゆる、CPUステッピング変更により、頻繁にBIOS版数をアップデートする必要があるため、BIOSをファームウェアでバインドせず、CPUバインドとする必要がある。即ち、CPUステッピング変更が起こった場合には、実装されているCPUに対応したBIOSより新しいものである必要がある。

【 0 0 0 8 】

従来技術では、BIOSのアップデートのための書込み中に、停電等により、BIOSを格納するメモリに破壊が生じた場合に、システム動作が不能となるといおそれがある。即ち、BIOSを格納するメモリからBIOSが、主記憶にロードされても、そのメモリのBIOSは、現在稼動中のため、停電等が生じると、書込み途中では、書込み前のBIOSが失われ、システム動作が不能となる。

【0009】

又、書込みが成功しても、復電時に、前のBIOSと異なるBIOSで復電処理することになり、これを考慮して、BIOSを変更する必要があるという制約が付けられ、BIOSのアップデートの範囲が限られる。

【0010】

従って、本発明の目的は、システムの稼動中に、BIOSの書き換えを行っても、システム起動が不能となることを防止するためのBIOSの冗長管理方法、データ処理装置及びストレージシステムを提供するにある。

【0011】

又、本発明の他の目的は、BIOSの書換えに失敗しても、システム起動が不能となることを防止するためのBIOSの冗長管理方法、データ処理装置及びストレージシステムを提供するにある。

【0012】

更に、本発明の他の目的は、BIOSの書換え中に停電が生じて、システムの復電処理を可能とするBIOSの冗長管理方法、データ処理装置及びストレージシステムを提供するにある。

【0013】**【課題を解決するための手段】**

この目的の達成のため、本発明のBIOSの冗長管理方法は、OSがハードウェアを使用できる環境に、前記ハードウェアを設定するBIOSを、各々格納する一対のメモリの一方を稼動中、他方を待機に使用するステップと、前記一方のメモリのBIOSがブート不能である時に、前記待機中のメモリのBIOSに切り換えるステップと、前記BIOSのアップデートを前記待機中のメモリへの書込みにより実行するステップとを有する。

【0014】

又、本発明のデータ処理装置は、CPUを含むハードウェアと、OSが前記ハードウェアを使用できる環境に、前記ハードウェアを設定するBIOSを、各々格納する一対のメモリと、前記ハードウェアの起動時に、前記一対のメモリの一方を稼動中、他方を待機に使用し、前記一方のメモリのBIOSがブート不能で

ある時に、前記待機中のメモリのBIOSに切り換えるサービスプロセッサとを有し、前記CPUは、前記BIOSのアップデートを前記待機中のメモリへの書込みにより実行する。

【0015】

又、本発明のストレージシステムは、CPUを含むハードウェアと、OSが前記ハードウェアを使用できる環境に、前記ハードウェアを設定するBIOSを、各々格納する一対のメモリと、前記ハードウェアの起動時に、前記一対のメモリ的一方を稼動中、他方を待機に使用し、前記一方のメモリのBIOSがブート不能である時に、前記待機中のメモリのBIOSに切り換えるサービスプロセッサとを有するストレージ制御装置と、前記ストレージ制御装置に接続された複数のストレージ装置とを有し、前記ストレージ制御装置の前記CPUは、前記BIOSのアップデートを前記待機中のメモリへの書込みにより実行する。

【0016】

本発明では、BIOSを一対のメモリで冗長管理し、BIOSのブート不能時に、待機中メモリに切り換え、システム起動不能を防止するとともに、CPUステッピング変更に応じたBIOSアップデート時に、同時に2つのBIOSメモリに、ライトせず、待機側のみにライトし、現在稼動中のBIOSを書き換えなため、アップデートに失敗した時に、現在稼動中のBIOSで起動できるため、システム起動不能になるのを防ぐことができる。

【0017】

又、待機側のメモリのみをアップデートするため、アップデートのライト中に停電が発生した場合、停電前と異なるBIOSで復電処理を行うことを防止できる。

【0018】

更に、本発明では、好ましくは、前記待機中のメモリへの前記BIOSのアップデートが成功した時に、前記待機中のメモリを前記稼動中に切り換えることを許可するステップを更に有する。これにより、アップデートしたBIOSに切り換えることを保証できる。

【0019】

更に、本発明では、好ましくは、前記ハードウェアの起動時に、前記許可された待機中のメモリを稼動中に、前記稼動中のメモリを前記待機中に切り換えるステップを更に有する。これにより、アップデートしたBIOSに自動切換えできる。

【0020】

更に、本発明では、好ましくは、前記切り換え後、前記稼動中に切り換えられたメモリのBIOSを、前記待機中に切り換えられたメモリに、書込み、冗長化するステップを更に有する。これにより、アップデートされてない他のメモリのBIOSもアップデートできる。

【0021】

更に、本発明では、好ましくは、前記待機中のメモリへの前記BIOSのアップデートが失敗した時に、前記待機中のメモリを前記稼動中に切り換えることを防止するステップを更に有する。これにより、アップデートが失敗したBIOSへの自動切換えを防止でき、不要な切り換えを防止できる。

【0022】

更に、本発明では、好ましくは、前記待機中に切り換えられたメモリへの前記BIOSの書込みが失敗した時に、前記待機中に切り替えられたメモリを前記稼動中に切り換えることを防止するステップを更に有する。これにより、冗長化に失敗したBIOSへの自動切換えを防止でき、不要な切り換えを防止できる。

【0023】

更に、本発明では、好ましくは、前記ハードウェアの前記待機中のメモリのBIOSアップデートに応じて、前記ハードウェアと接続された他のハードウェアの待機中のメモリのBIOSアップデートを実行するステップを更に有する。これにより、一対のハードウェアのBIOSアップデートを同時にできる。

【0024】

更に、本発明では、好ましくは、前記ハードウェアと接続された他のハードウェアとのBIOSの同期化処理を行うステップを更に有する。これにより、ハードウェア間のBIOS版数を一致することができる。

【0025】

【発明の実施の形態】

以下、本発明の実施の形態を、ストレージシステム、BIOSの冗長管理処理、CM間のBIOS同期化処理、他の実施の形態の順で説明する。

【0026】**[ストレージシステム]**

図1は、本発明の一実施の形態のストレージシステムの構成図であり、磁気ディスクを使用したRAID (Redundant Arrays of Inexpensive Disk)システムを示す。図1に示すように、ストレージシステムは、一对の磁気ディスクコントローラ（以下、コントローラという）1、2と、この一对のコントローラ1、2にライン11、12で接続された多数の磁気ディスク装置50-1～50-m、52-1～52-nとからなる。

【0027】

コントローラ1、2は、直接又はネットワーク機器を介し、ホストやサーバーに接続され、ホストやサーバーの大量のデータを、RAIDディスクドライブ（磁気ディスク装置）へ高速かつ、ランダムに読み書きが出来るシステムである。一对のコントローラ1、2は、同一の構成を有し、CA (Channel Adapter) 11、12、21、22と、CM (Centralized Module) 10、15～19、20、25～29と、DA (Device Adapter) 13、14、23、24のファンクションモジュールによって構成されている。

【0028】

CA (Channel Adapter) 11、12、21、22は、ホストを結ぶホスト・インタフェースの制御をつかさどる回路であり、例えば、ファイバーチャネル回路 (FC) とDMA (Direct Memory Access) 回路等で構成される。DA (Device Adapter) 13、14、23、24は、ディスクデバイス50-1～50-m、52-1～52-mを制御するため、ディスクデバイスとコマンド、データのやり取りを行う回路であり、例えば、ファイバーチャネル回路 (FC) とDMA回路等で構成される。

【0029】

CM (Centralized Module) は、CPU 10、20と、ブリッジ回路17、2

7と、メモリ(RAM)15、25と、コンパクトフラッシュメモリ16、26と、IOブリッジ回路18、28と、一対のBIOSフラッシュメモリ32、33、42、43とを有する。更に、CMは、RSP(Remote Service Processor)34、44と、外部接続用LANポート36、46とを有する。メモリ15、25は、バッテリーでバックアップされ、主記憶として使用される。

【0030】

CPU10、20は、ブリッジ回路17、27を介し、メモリ15、25、コンパクトフラッシュメモリ16、26、IOブリッジ回路18、28に接続される。このメモリ15、25は、CPU10、20のワーク領域に使用され、コンパクトフラッシュメモリ16、26は、CPU10、20が実行するプログラムを格納する。このプログラムとして、カーネル、ファイルアクセスプログラム(リード/ライトプログラム)、RAID管理プログラム等を格納する。

【0031】

BIOSフラッシュメモリ32、33、42、43は、冗長構成のため、一対設けられ、一方が稼動、他方が待機に使用され、BIOS(図4で後述)を格納する。CPU10、20は、このプログラムを実行し、リード/ライト処理、RAID管理処理等を実行する。

【0032】

PCIバス35、45は、ブリッジ回路17、27を介し、CPU10、20と、コンパクトフラッシュメモリ15、25、一対のBIOSフラッシュメモリ32、33、42、43、RSP34、44、LANポート36、46とを接続する。

【0033】

RSP34、44は、各種のリモートサービスを行うプロセッサで構成され、本実施例では、BIOSフラッシュメモリ32、33、42、43の冗長管理を行う。LANポート36、46は、外部のLAN(Local Area Network)と接続するためのものである。

【0034】

PCI(Personal Computer Interface)バス31、41は、CA11、12

、21、22と、DA13、14、23、24とを接続するとともに、IOブリッジ回路18、28を介し、CPU10、20、メモリ15、25を接続する。更に、PCIバス31、41には、PCI-ノードリンクブリッジ(PNB)回路30、40が接続される。

【0035】

コントローラ1のPCI-ノードリンクブリッジ回路30は、コントローラ2のPCI-ノードリンクブリッジ回路40と接続され、コントローラ1、2間のコマンド、データの交信を行う。

【0036】

コントローラ1は、例えば、ディスク装置50-1～50-mを担当し、コントローラ2は、例えば、ディスク装置52-1～52-nを担当する。図1では、ディスク装置50-1～50-mと、52-1～52-nとが、RAID5の構成を有する。

【0037】

図2は、図1のコンパクトフラッシュメモリ16、26に格納されたプログラムの一例であり、カーネル102、システム制御104、パワー制御106、構成管理108、保守タスク110、フラッシュドライバ112、RSPドライバ114等で構成される。カーネル102は、OSであり、カーネル102以外は、ファームウェアである。

【0038】

図3は、図1のRSP34、44のNVRAM（不揮発性ランダムアクセスメモリ）に格納されたBIOS冗長管理情報の説明図であり、boot model20は、BIOSのブートモード（FAST/SLOW）を格納する。Current model22は、現在稼動中のBIOS番号#を格納する。BIOS SW124は、次回起動時に起動するBIOS番号#を格納する。待機BIOS版数126は、待機側のBIOS版数を格納する。

【0039】

図4は、図1のBIOSフラッシュメモリに格納されるBIOSの処理フロー図である。BIOSは、前述のように、OS（カーネル）が使用するハードウェア

アをチェックし、OS（カーネル）が、ハードウェアを使用できる環境に設定する。従って、OSのロード前に行われる。

【0040】

(S10) RSP34, 44が起動するBIOSを設定し、先ず、CPU10, 20のリセットが解除されると、CPU10, 20が、BIOSフラッシュメモリ32（又は33）、42（又は43）のBIOSの先頭ブロックを読み出し、RSP34, 44の初期化、即ち、BIOSがRSP34, 44の機能を使用できるような設定を、BIOS Boot blockの先頭で行う。次に、CPU10, 20の初期化を行う。即ち、CPU10, 20を使用できるように、レジスタの設定、マシンチェックの初期化等を行う。

【0041】

(S12) 各チップセット（各ブリッジ回路17, 18, 27, 28等）の初期化（Disable, レジスタ設定等）を行う。又、メモリ15, 25の初期化（使用可能な状態にし、診断及びECCチェック等）を行う。

【0042】

(S14) このメモリ15, 25、チップセットの初期化の後、BIOSフラッシュメモリからメモリ15, 25に、BIOSをロードする。次に、PCIバス31, 35, 41, 45に接続されたPCIデバイス（CA11, 12, 21, 22, DA13, 14, 23, 24, LANポート36, 46）の初期化を行う。

【0043】

(S16) 更に、必要に応じて、他のデバイスの初期化を行う。

【0044】

(S18) 次に、各種のテーブルを作成し、ブートを終了する。これにより、コンパクトフラッシュメモリ16, 26からカーネル等がメモリ15, 25にロードされ、プログラムが起動される。

【0045】

図1のストレージシステムにおいて、コントローラ1, 2では、メモリ15, 25に配置されたキャッシュメモリは、各々、担当するディスク装置のデータの

一部を格納し、ホストからのライトデータを格納する。CPU10, 20は、CA11, 12, 21, 22を介しホストからのリード要求を受けて、キャッシュメモリを参照し、物理ディスクへのアクセスが必要かを判定し、必要であれば、ディスクアクセス要求をDA13, 14, 23, 24に要求する。又、CPU10, 20は、ホストからのライト要求を受けて、ライトデータをキャッシュメモリに書き込み、且つ内部でスケジュールされるライトバック等をDA13, 14, 23, 24に要求する。

【0046】

[BIOSの冗長管理処理]

前述のように、各コントローラ1, 2に、BIOSフラッシュメモリ (Flash ROM) を物理的に2つ実装している。この2つのフラッシュメモリへ同じ版数のBIOSを格納し、片方のBIOSフラッシュメモリ (Flash ROM) 32, 42がBoot不能になっても、もう一方33, 43から同じ版数のBIOSを起動することができるように、冗長管理 (図8で後述) を行う。

【0047】

このBIOSの冗長化は、BIOS処理が終了後に、バイオス冗長処理ファームウェア (図9で後述) で行う。又、起動するBIOSは、RSP34, 44のProcessorにより、切り替えを行う。

【0048】

先ず、BIOSのアップデートを、図5、図6、図7、図10乃至図12で説明する。図5に示すように、BIOSフラッシュメモリ (Flash ROM) 33へのフラッシュライトは、ユーザインタフェイスを使用して、ファームウェアから行う。

【0049】

即ち、コントローラ1のLANポート36に、ハブ7を介しパーソナルコンピュータ (以下、PCという) 6を接続し、図6及び図7の処理により、実行する。図6は、BIOSアップデート指示処理フロー図である。

【0050】

(S20) PC6のCGI画面より、BIOSアップデート指示を行う。即ち

、BIOSアップデート画面を表示し、アップデートを指示する。

【0051】

(S21) コントローラ1のCPU10が実行する保守タスク110は、構成からBIOS版数を獲得し、PC6のCGIに通知する。

【0052】

(S22) PC6のCGIでは、通知された現在稼動中のBIOS版数をCGI画面に表示する。そして、ユーザが確認後、PC6のCGIは、BIOS ROM Imageを、CPU10が実行する保守タスク110に転送する。

【0053】

(S23) 保守タスク110は、CGIから受け取ったBIOS ROM Imageのチェックサムをチェックし、異常だったら、CGIに異常を通知する。異常でなければ、転送されたBIOSの版数を、CGIに通知する。

【0054】

(S24) CGIでは、CGI画面に、チェックサムエラーが発生した場合は、その旨を表示する。

【0055】

(S25) CGIでは、正常だった場合は、保守タスクから受け取ったBIOS版数を、画面に表示し、このBIOSにアップデートしていいか最終確認を行う。

【0056】

(S26) 続行なら、CGIは、フラッシュライト指示を保守タスク110へ送る。保守タスク110は、フラッシュライト指示を受け、図7のBIOSフラッシュライト処理を実行する。

【0057】

図7は、保守タスクが実行するフラッシュライト処理フロー図である。

【0058】

(S30) 保守タスク110は、BIOSフラッシュライト指示を受けると、RSP34, 44内のNVRAM (図3参照) のCurrent SW122から現在稼動中のBIOSのフラッシュメモリ番号 (図1の32か33) を獲得する。

【0059】

(S32) 次に、保守タスク110は、RSP34, 44内のNVRAM (図3参照) の待機BIOS版数126の版数を無効化する。これにより、BIOSフラッシュROMの自動切換えを防止する。

【0060】

(S34) ステップS30の現在稼動中のBIOSのフラッシュメモリ番号から待機中のBIOSフラッシュメモリを求め、カーネル102が用意する関数を使用し、現在稼動中ではない方(待機側)のBIOSフラッシュROMに、転送されたBIOSをフラッシュライトする。このとき、BIOS Boot Block部もフラッシュライトする。

【0061】

(S36) 保守タスク110は、正常にフラッシュライト終了したかを判定する。

【0062】

(S38) 保守タスク110は、正常にフラッシュライト終了したと判定すると、RSP34, 44内のNVRAM (図3参照) のBIOS SW124の次回起動時に起動するBIOS番号に、フラッシュライトしたBIOSフラッシュROM番号を設定する。又、保守タスク110は、RSP34, 44内のNVRAM (図3参照) の待機BIOS版数126に、このフラッシュライトされたBIOS版数を設定し、この書き換えられたフラッシュROMを有効化する。従って、次の起動時には、書き換えられたBIOSが選択される。更に、WebにBIOSのアップデートが正常終了したことを通知し、PC6のCGIで確認させる。そして、終了する。

【0063】

(S40) 一方、ステップS36で、BIOSアップデート時に、フラッシュライトエラーが発生したことを検出した場合には、待機側BIOSフラッシュメモリが異常であることを、システム制御104に通知し、エラーが発生したコントローラの状態を予防保守が必要な状態にする(例えば、状態ランプはOrangeとする)。そして、PC6のCGI画面に、BIOSアップデートが失敗したこと

が通知される。この場合に、BIOS SW124、待機BIOS版数126は、更新されないため、エラーが発生したBIOSフラッシュROMへの自動切り換えを防止できる。

【0064】

このように、BIOSアップデート時に、同時に2つのBIOSフラッシュROMに、フラッシュライトせず、待機側のみにフラッシュライトする。この理由は、現在稼働中のBIOSを書き換えるのは危険であるからである。即ち、フラッシュライトに失敗したときに、現在稼働中のBIOSを書き換えないため、現在稼働中のBIOSで起動できるため、システム起動不能になるのを防ぐことができる。

【0065】

又、フラッシュライト中に停電が発生した場合、停電前と異なるBIOSで復電処理を行うことになり、版数の異なるBIOS間でFast Bootを保障しなければならないため、待機側のみにフラッシュライトする。尚、Fast Bootとは、停電発生時に、コントローラ内のメモリ15、25のキャッシュ域のデータは、バッテリーバックアップにより保持されるため、復電時に、キャッシュ上のデータを保証するため、メモリ初期化を省略して、コントローラを立ち上げるモードである。ここで、停電／復電を挟んで、BIOS版数が異なると、ハード初期化手順が相違し、メモリデータが保証できなくなる。

【0066】

更に、図1の2つのコントローラ1、2が接続されたモデルでは、図1のコントローラ1の保守タスク110と同様に、コントローラ1がPC6から受けたコマンド、情報が、PNB30、40経由で、コントローラ2に伝達され、コントローラ2の保守タスク110が同一動作を行う。従って、コントローラ2の待機側BIOSフラッシュROMも同時にアップデートされる。

【0067】

この場合に、BIOSアップデート時に、片方のコントローラだけBIOSアップデートが失敗した場合には、PC6のCGI画面に、BIOSアップデートが失敗したことが通知される。アップデートに失敗したコントローラの次回起動

時に立ち上げるBIOSは切換えできず、現在のBIOSで起動する。この状態では、2つのコントローラのBIOSの冗長化は行なわれていないが、次のパワーオン時に、図13で後述するように、冗長化される。

【0068】

次に、コントローラ起動時のBIOSの切り換え処理を説明する。図8は、コントローラ起動時のRSPのBIOS起動処理フロー図である。

【0069】

(S50) 適当なタイミングでコントローラが再起動されると、RSP34、44は、RSP34、44内のNVRAM(図3参照)のboot mode120からBoot mode(Fast/Slow)を獲得する。Fast boot modeは、停電後の復電などで前回起動したBIOSで起動するモードである。一方、Slow boot modeは、通常のパワーオンなどで書き換えられたBIOSで起動するモードである。

【0070】

(S52) RSP34、44は、boot modeを判定し、FASTなら、ステップS56に進む。即ち、ステップS54をジャンプし、停電前のBIOSを使用し、復電時の整合性を取る。

【0071】

(S54) 一方、Slowと判定すると、RSP34、44内のNVRAM(図3参照)のBIOS SW124の次回起動時に起動するBIOS番号を獲得し、RSP34、44内のNVRAM(図3参照)のCurrent SW122に、獲得したBIOS番号を設定する。従って、起動BIOSは、書き換えられたBIOSに切り替えられる。

【0072】

(S56) RSP34、44は、NVRAM(図3参照)のCurrent SW122に設定されたBIOSを起動する。

【0073】

このようにして、起動時に、復電時以外は、更新されたBIOSに切り替わる。復電時は、前のBIOSを起動する。

【0074】

図9は、パワー制御が実行するBIOS冗長化処理フロー図である。

【0075】

(S60) BIOS処理(図4)が終了すると、パワー制御106内のBIOS冗長化処理は、RSP34, 44内のNVRAM(図3参照)のBIOS SW124の次回起動時に起動するBIOS番号と、Current SW122の現在稼働中のBIOS番号を獲得する。

【0076】

(S62) BIOS SW124の次回起動時に起動するBIOS番号と、Current SW122の現在稼働中のBIOS番号とが一致しているかを判定する。一致していない場合には、停電後の復電時等、次回に指定したBIOSで起動していないため、冗長化処理を行わず、終了する。

【0077】

(S64) BIOS番号が一致している場合には、RSP34, 44の待機BIOS版数126が無効かをチェックする。待機BIOS版数が無効である場合には、待機BIOSが異常なため、ステップS68の冗長化に進む。

【0078】

(S66) 待機BIOSが無効でない場合には、稼働中と待機中の両BIOSの版数を比較し、一致するかを判定する。一致する場合には、両BIOSの版数が同じであるから、冗長化は必要なく、終了する。

【0079】

(S68) 一致しない場合には、冗長化が必要であり、稼働中BIOS Imageを、待機側のBIOSフラッシュメモリにフラッシュライトする。このときに、ライトするBIOS ROM Imageは、現在稼働中のBIOSフラッシュROMのデータを使用する。そして、RSP34, 44のNVRAMの待機BIOS版数を設定し、待機側を有効化し、冗長化処理を終了する。

【0080】

図10乃至図12は、その動作説明図である。図10に示すように、図6の処理により、メモリ15、25に、転送BIOSが書き込まれる。図6でフラッシ

ユライトが許可されると、図11に示すように、図7の処理で、メモリ15、25の転送BIOSが、待機側のBIOSフラッシュROM32、42に書き込まれる。そして、図12に示すように、コントローラの起動時には、図8の処理により、待機側が起動され、稼動中のBIOSフラッシュROM33、43は、待機側となる。又、図9の処理により、稼動中に変更されたBIOSフラッシュROM32、42のBIOSが、待機中に変更されたBIOSフラッシュROM33、43に書き込まれる。

【0081】

又、ステップS68のBIOS冗長化時に、フラッシュライトエラーが発生した場合には、コントローラの動作自体には問題がないため、レディで起動するが、エラーが発生したコントローラの状態を予防保守が必要な状態にする（状態ランプはOrange）。更に、エラーが発生したBIOSフラッシュROMへの自動切り換えを行なわないように、待機側を有効化しない。

【0082】

更に、BIOSアップデート後の再起動で、BIOSが正常に起動しない場合（新BIOSが正常に起動しなかった場合）には、自動でBIOSフラッシュROMを切り換えることはせず、フロントパネルを使用して古い版数のBIOSに切り替えを行う。起動しないことで、ユーザは、新しいBIOSが正常に起動しなかったことがわかる。

【0083】

更に、通常の運用においては、2つのBIOSフラッシュROMの片方のフラッシュROMが異常になった場合でも、もう一方のフラッシュROMでBIOSが起動できる。この切り替え方法として、ユーザインタフェイスからの指示により、BIOSフラッシュROMを切り換える。

【0084】

又、自動切り換えの方法としては、RSP34、44が、BIOSのBoot block 処理中にHeart Beat Error（BIOSからの応答なし）を検出し、BIOSフラッシュROMを切り替える。ただし、BIOSフラッシュROMを切り替えるのは、待機側のBIOSが使用可能（有効）である場合のみで、使用不可

能な場合は、切替えを行わず、デグレードさせる。切り替え発生後、BIOS処理が終了し、ファームウェアまで起動することができれば、図9で説明したBIOS冗長化処理が実行される。

【0085】

[CM間のBIOS同期化処理]

次に、図1のように、2つのコントローラを搭載した場合に、図1に示したCM (Centralized Module) 間のBIOS同期化処理を説明する。例えば、図14に示すように、コントローラ2のCM2が、障害を起こし、CM2' に交換した場合に、コントローラ1のCM1とコントローラ2のCM2' とで、BIOSの同期化を行い、CM交換時のBIOS版数の食い違いをなくす。

【0086】

Slave CM (例えば、コントローラ2のCM) のBIOSをアップデートした場合は、自動で再起動し、新しいBIOSで起動する。Master CM (例えば、コントローラ1のCM) のBIOSをアップデートした場合は、自動Rebootを行わず、ユーザにこのことを通知する。

【0087】

図13は、このCM間のBIOS同期処理フロー図である。

【0088】

(S70) マスタCMが、BIOS同期化処理を開始する。まず、スレーブCMのBIOS版数を獲得する。

【0089】

(S72) マスタCMは、マスタCMのBIOS版数とスレーブCMのBIOS版数とを比較する。一致する場合には、BIOS同期化が不要のため、終了する。

【0090】

(S74) 比較により、マスタCMのBIOS版数がスレーブCMのBIOS版数より小さい、即ち、マスタCMのBIOSが古い場合には、マスタCMのBIOSアップデートが必要である。まず、マスタCMは、スレーブCMに、BIOSデータの転送を要求する。

【0091】

(S76) スレーブCMでは、スレーブCMの稼動中のBIOSフラッシュROMからBIOSをリードし、マスタCMへ転送する。

【0092】

(S78) マスタCMでは、転送されたBIOSを待機側のBIOSフラッシュROMにライトする。これにより、古い版数のBIOSがアップデートされる。

【0093】

(S80) マスタCMは、BIOSのライトが成功したかを判定し、成功した場合には、ユーザに、BIOSがアップデートされ、再起動が必要なことを通知する。従って、再起動により、BIOSアップデートが完了する。逆に、BIOSライトが失敗した場合には、マスタCMの待機側BIOSフラッシュROMが、異常なため、マスタCMを予防保守対象にし、異常をユーザに通知する。

【0094】

(S82) 逆に、ステップS72の比較により、マスタCMのBIOS版数がスレーブCMのBIOS版数より大きい、即ち、マスタCMのBIOSが新しい場合には、スレーブCMのBIOSアップデートが必要である。先ず、マスタCMは、マスタCMの稼動中のBIOSフラッシュROMからBIOSをリードし、スレーブCMへ転送する。

【0095】

(S84) スレーブCMでは、転送されたBIOSを待機側のBIOSフラッシュROMにライトする。これにより、古い版数のBIOSがアップデートされる。更に、ライト結果をマスタCMに通知する。

【0096】

(S86) マスタCMは、通知結果からBIOSのライトが成功したかを判定し、成功した場合には、スレーブCMが再起動され、BIOSアップデートが完了する。逆に、BIOSライトが失敗した場合には、スレーブマスタCMの待機側BIOSフラッシュROMが、異常なため、スレーブCMを予防保守対象にし、異常をユーザに通知する。

【0097】

このようにして、CM間で新しいBIOSに同期化される。

【0098】

[他の実施の形態]

前述の実施の形態では、図1のような冗長構成のRAIDで説明したが、これ以外の冗長構成のストレージシステムに適用できる。又、物理ディスクは、磁気ディスク、光ディスク、光磁気ディスク、各種のストレージデバイスを適用できる。

【0099】

又、ストレージシステムの適用を説明したが、ストレージに限らず、他のコントローラやデータ処理装置に適用できる。更に、2つのCMの例で説明したが、1つのCMでも適用でき、BIOSの格納に、フラッシュメモリを使用したか、他の不揮発性な書き換え可能なメモリを使用できる。

【0100】

以上、本発明を実施の形態により説明したが、本発明の趣旨の範囲内において、本発明は、種々の変形が可能であり、本発明の範囲からこれらを排除するものではない。

【0101】

(付記1) OSがハードウェアを使用できる環境に、前記ハードウェアを設定するBIOSを、各々格納する一対のメモリの一方を稼動中、他方を待機に使用するステップと、前記一方のメモリのBIOSがブート不能である時に、前記待機中のメモリのBIOSに切り換えるステップと、前記BIOSのアップデートを前記待機中のメモリへの書込みにより実行するステップとを有することを特徴とするBIOSの冗長管理方法。

【0102】

(付記2) 前記待機中のメモリへの前記BIOSのアップデートが成功した時に、前記待機中のメモリを前記稼動中に切り換えることを許可するステップを更に有することを特徴とする付記1のBIOSの冗長管理方法。

【0103】

(付記 3) 前記ハードウェアの起動時に、前記許可された待機中のメモリを稼動中に、前記稼動中のメモリを前記待機中に切り換えるステップを更に有することを特徴とする付記 2 の BIOS の冗長管理方法。

【0104】

(付記 4) 前記切り換え後、前記稼動中に切り換えられたメモリの BIOS を、前記待機中に切り換えられたメモリに、書込み、冗長化するステップを更に有することを特徴とする付記 3 の BIOS の冗長管理方法。

【0105】

(付記 5) 前記待機中のメモリへの前記 BIOS のアップデートが失敗した時に、前記待機中のメモリを前記稼動中に切り換えることを防止するステップを更に有することを特徴とする付記 1 の BIOS の冗長管理方法。

【0106】

(付記 6) 前記待機中に切り換えられたメモリへの前記 BIOS の書込みが失敗した時に、前記待機中に切り替えられたメモリを前記稼動中に切り換えることを防止するステップを更に有することを特徴とする付記 4 の BIOS の冗長管理方法。

【0107】

(付記 7) 前記ハードウェアの起動が復電である場合には、前記切り換えの実行を防止することを特徴とする付記 3 の BIOS の冗長管理方法。

【0108】

(付記 8) 前記ハードウェアの起動が復電である場合には、前記冗長化の実行を防止することを特徴とする付記 4 の BIOS の冗長管理方法。

【0109】

(付記 9) 前記ハードウェアの前記待機中のメモリの BIOS アップデートに応じて、前記ハードウェアと接続された他のハードウェアの待機中のメモリの BIOS アップデートを実行するステップを更に有することを特徴とする付記 1 の BIOS の冗長管理方法。

【0110】

(付記 10) 前記ハードウェアと接続された他のハードウェアとの BIOS の

同期化処理を行うステップを更に有することを特徴とする付記1のBIOSの冗長管理方法。

【0111】

(付記11) CPUを含むハードウェアと、OSが前記ハードウェアを使用できる環境に、前記ハードウェアを設定するBIOSを、各々格納する一対のメモリと、前記ハードウェアの起動時に、前記一対のメモリ的一方を稼動中、他方を待機に使用し、前記一方のメモリのBIOSがブート不能である時に、前記待機中のメモリのBIOSに切り換えるサービスプロセッサとを有し、前記CPUは、前記BIOSのアップデートを前記待機中のメモリへの書込みにより実行することを特徴とするデータ処理装置。

【0112】

(付記12) 前記サービスプロセッサは、前記待機中のメモリへの前記BIOSのアップデートが成功した時に、前記待機中のメモリを前記稼動中に切り換えることを許可することを特徴とする付記11のデータ処理装置。

【0113】

(付記13) 前記サービスプロセッサは、前記ハードウェアの起動時に、前記許可された待機中のメモリを稼動中に、前記稼動中のメモリを前記待機中に切り換えることを特徴とする付記12のデータ処理装置。

【0114】

(付記14) 前記CPUは、前記切り換え後、前記稼動中に切り換えられたメモリのBIOSを、前記待機中に切り換えられたメモリに、書込み、冗長化することを特徴とする付記13のデータ処理装置。

【0115】

(付記15) 前記CPUは、前記待機中のメモリへの前記BIOSのアップデートが失敗した時に、前記待機中のメモリを前記稼動中に切り換えることを防止することを特徴とする付記11のデータ処理装置。

【0116】

(付記16) 前記CPUは、前記待機中に切り換えられたメモリへの前記BIOSの書込みが失敗した時に、前記待機中に切り換えられたメモリを前記稼動中

に切り換えることを防止することを特徴とする付記 14 のデータ処理装置。

【0117】

(付記 17) 前記ハードウェアと接続された他のハードウェアを更に有し、前記ハードウェアの前記待機中のメモリの BIOS アップデートに応じて、前記ハードウェアと接続された他のハードウェアの待機中のメモリの BIOS アップデートを実行することを特徴とする付記 11 のデータ処理装置。

【0118】

(付記 18) 前記ハードウェアは、前記ハードウェアと接続された他のハードウェアとの BIOS の同期化処理を行うことを特徴とする付記 11 のデータ処理装置。

【0119】

(付記 19) CPU を含むハードウェアと、OS が前記ハードウェアを使用できる環境に、前記ハードウェアを設定する BIOS を、各々格納する一対のメモリと、前記ハードウェアの起動時に、前記一対のメモリの一方を稼動中、他方を待機に使用し、前記一方のメモリの BIOS がブート不能である時に、前記待機中のメモリの BIOS に切り換えるサービスプロセッサとを有するストレージ制御装置と、前記ストレージ制御装置に接続された複数のストレージ装置とを有し、前記ストレージ制御装置の前記 CPU は、前記 BIOS のアップデートを前記待機中のメモリへの書込みにより実行することを特徴とするストレージシステム。

【0120】

(付記 20) 前記ストレージ制御装置の前記サービスプロセッサは、前記待機中のメモリへの前記 BIOS のアップデートが成功した時に、前記待機中のメモリを前記稼動中に切り換えることを許可することを特徴とする付記 19 のストレージシステム。

【0121】

(付記 21) 前記ストレージ制御装置の前記サービスプロセッサは、前記ハードウェアの起動時に、前記許可された待機中のメモリを稼動中に、前記稼動中のメモリを前記待機中に切り換えることを特徴とする付記 20 のストレージシステム。

ム。

【0122】

(付記22) 前記ストレージ制御装置の前記CPUは、前記切り換え後、前記稼動中に切り換えられたメモリのBIOSを、前記待機中に切り換えられたメモリに、書込み、冗長化することを特徴とする付記21のストレージシステム。

【0123】

(付記23) 前記ストレージ制御装置の前記CPUは、前記待機中のメモリへの前記BIOSのアップデートが失敗した時に、前記待機中のメモリを前記稼動中に切り換えることを防止することを特徴とする付記19のストレージシステム。

【0124】

(付記24) 前記ストレージ制御装置の前記CPUは、前記待機中に切り換えられたメモリへの前記BIOSの書込みが失敗した時に、前記待機中に切り替えられたメモリを前記稼動中に切り換えることを防止することを特徴とする付記22のストレージシステム。

【0125】

(付記25) 前記ストレージ装置及び前記ストレージ制御装置に接続され、前記ストレージ装置を制御する他のストレージ制御装置を更に有し、前記ストレージ制御装置の前記待機中のメモリのBIOSアップデートに応じて、前記他のストレージ制御装置の待機中のメモリのBIOSアップデートを実行することを特徴とする付記19のストレージシステム。

【0126】

(付記26) 前記ストレージ装置及び前記ストレージ制御装置に接続され、前記ストレージ装置を制御する他のストレージ制御装置を更に有し、前記ストレージ制御装置は、前記他のストレージ制御装置とのBIOSの同期化処理を行うことを特徴とする付記19のストレージシステム。

【0127】

【発明の効果】

このように、本発明では、BIOSを一对のメモリで冗長管理し、BIOSの

ブート不能時に、待機中メモリに切り換え、システム起動不能を防止する。又、CPUステッピング変更に応じたBIOSアップデート時に、同時に2つのBIOSメモリに、ライトせず、待機側のみにライトし、現在稼動中のBIOSを書き換えなため、アップデートに失敗した時に、現在稼動中のBIOSで起動でき、システム起動不能になるのを防ぐことができる。

【0128】

又、待機側のメモリのみアップデートするため、アップデートのメモリライト中に停電が発生した場合、停電前と異なるBIOSで復電処理を行うことを防止できる。

【図面の簡単な説明】

【図1】

本発明の一実施の形態のストレージシステムの構成図である。

【図2】

図1の格納プログラムの構成図である。

【図3】

図1のRSPの冗長管理情報の説明図である。

【図4】

図1のBIOSの処理フロー図である。

【図5】

本発明の一実施の形態のBIOSアップデートの説明図である。

【図6】

図5のBIOSアップデート処理フロー図である。

【図7】

図6のBIOSフラッシュライト処理フロー図である。

【図8】

図1のCM起動時のRSPの処理フロー図である。

【図9】

図1のBIOS冗長化処理フロー図である。

【図10】

図 6 の B I O S アップデート処理の動作説明図である。

【図 1 1】

図 7 の B I O S フラッシュライト処理の動作説明図である。

【図 1 2】

図 9 の B I O S 冗長化処理の動作説明図である。

【図 1 3】

本発明の他の実施の形態の CM 間 B I O S 同期化処理フロー図である。

【図 1 4】

図 1 3 の CM 間 B I O S 同期化処理の動作説明図である。

【符号の説明】

1、2 ストレージコントローラ

7 ハブ

6 パーソナルコンピュータ

11、12、21、23 チャンネルアダプター

13、14、23、24 デバイスアダプター

10、20 CPU

15、25 メモリ

16、26 プログラムメモリ

32、33、42、43 B I O S フラッシュROM

34、44 RSP

30、40 PCI-ノードブリッジ回路

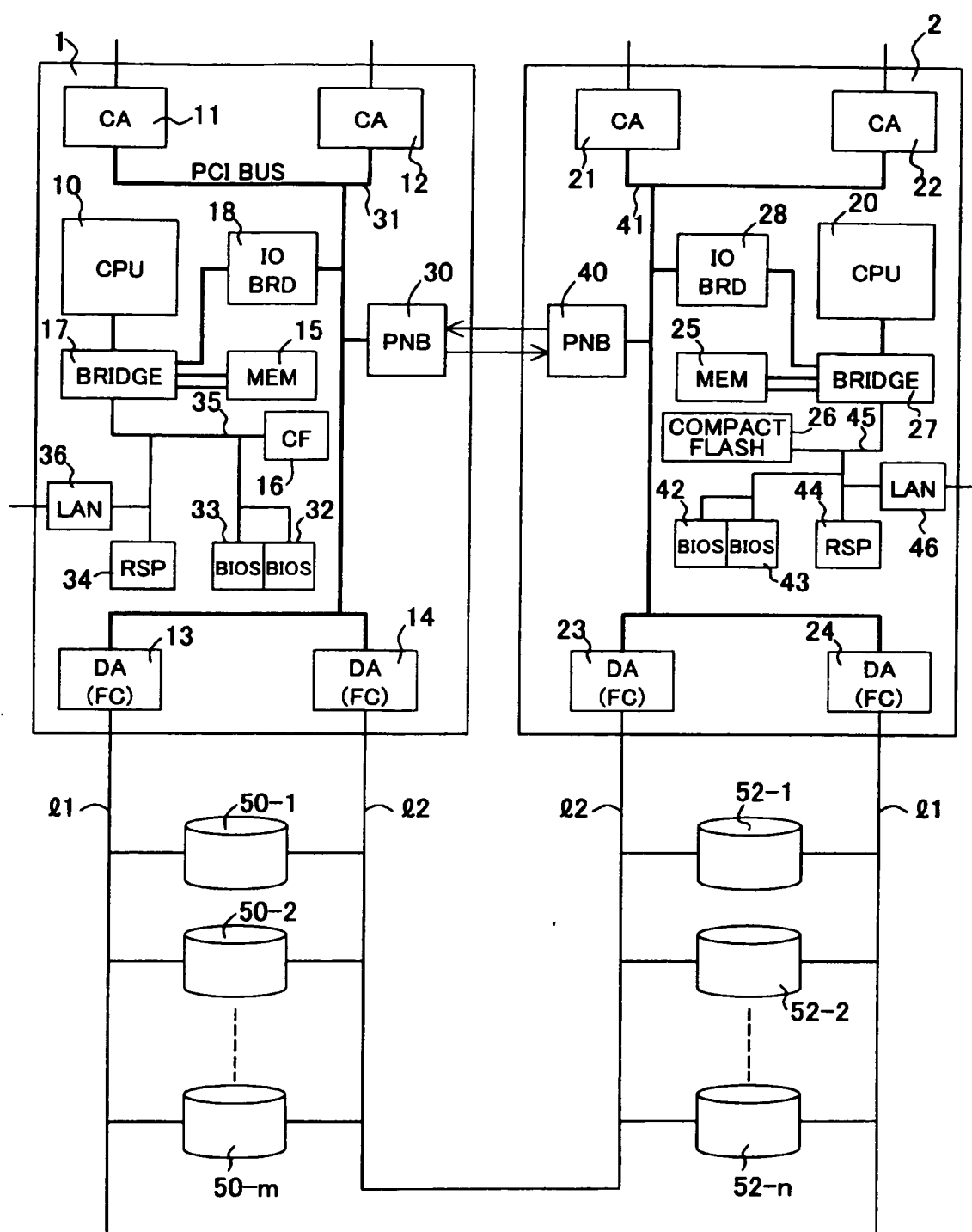
31、41 PCIバス

36、46 LANポート

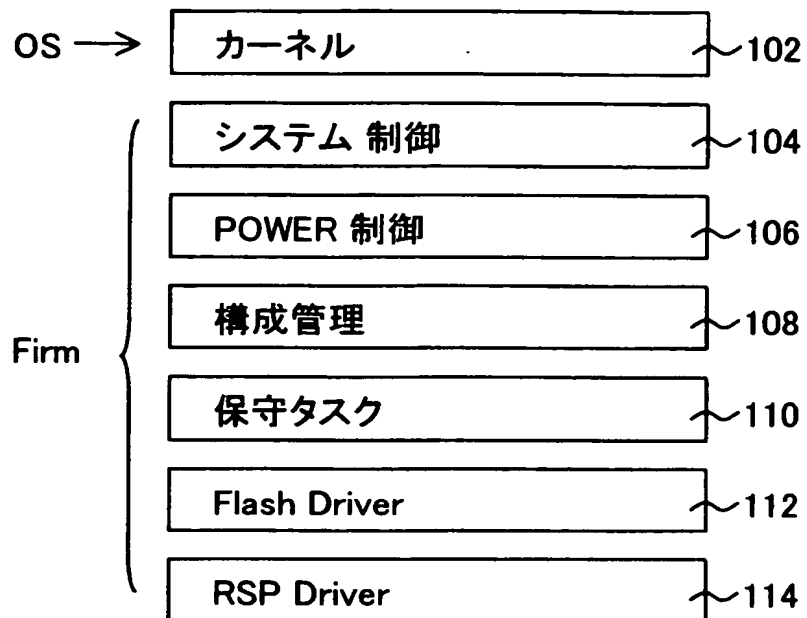
50-1～50-m、52-1～52-n 物理ディスク装置（ストレージ装置）

【書類名】 図面

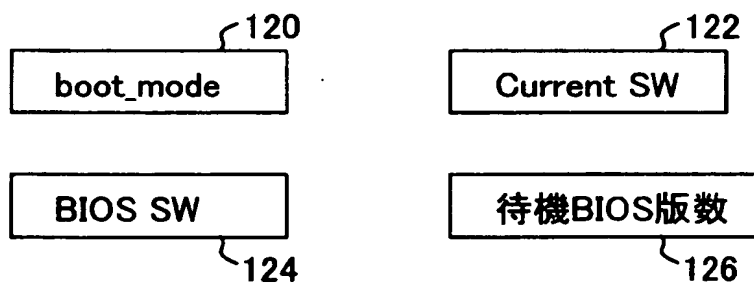
【図 1】



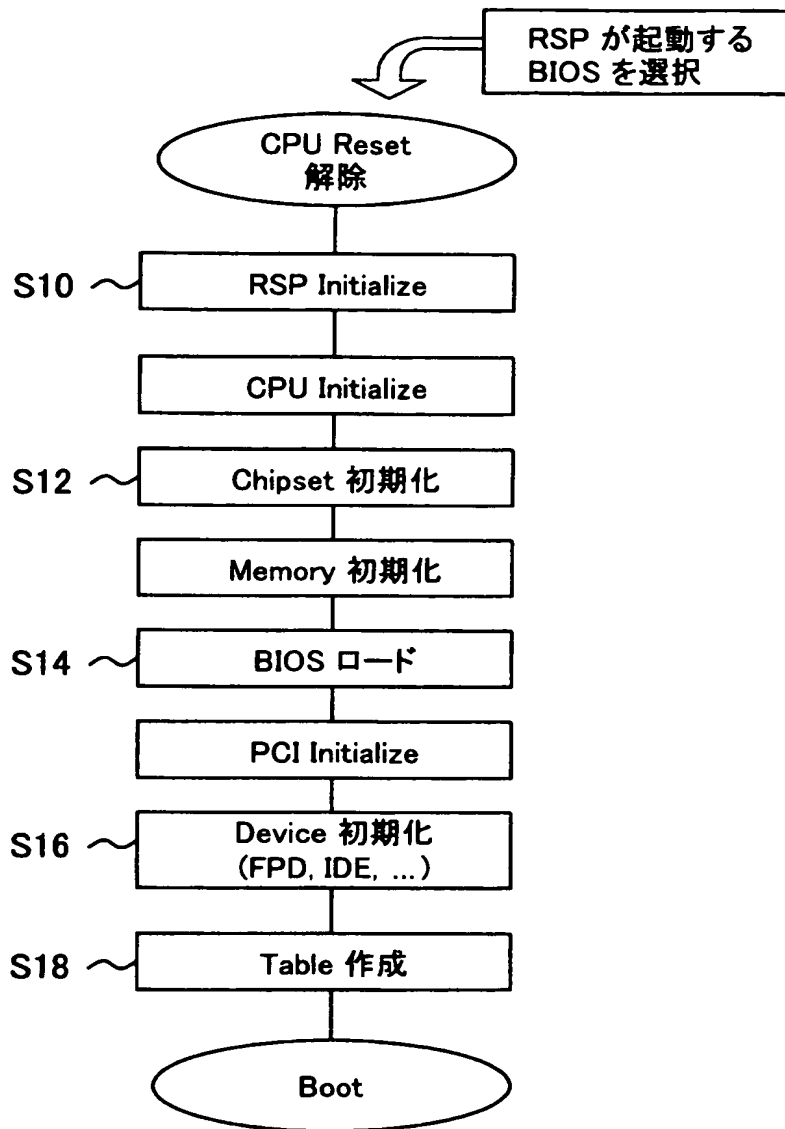
【図 2】



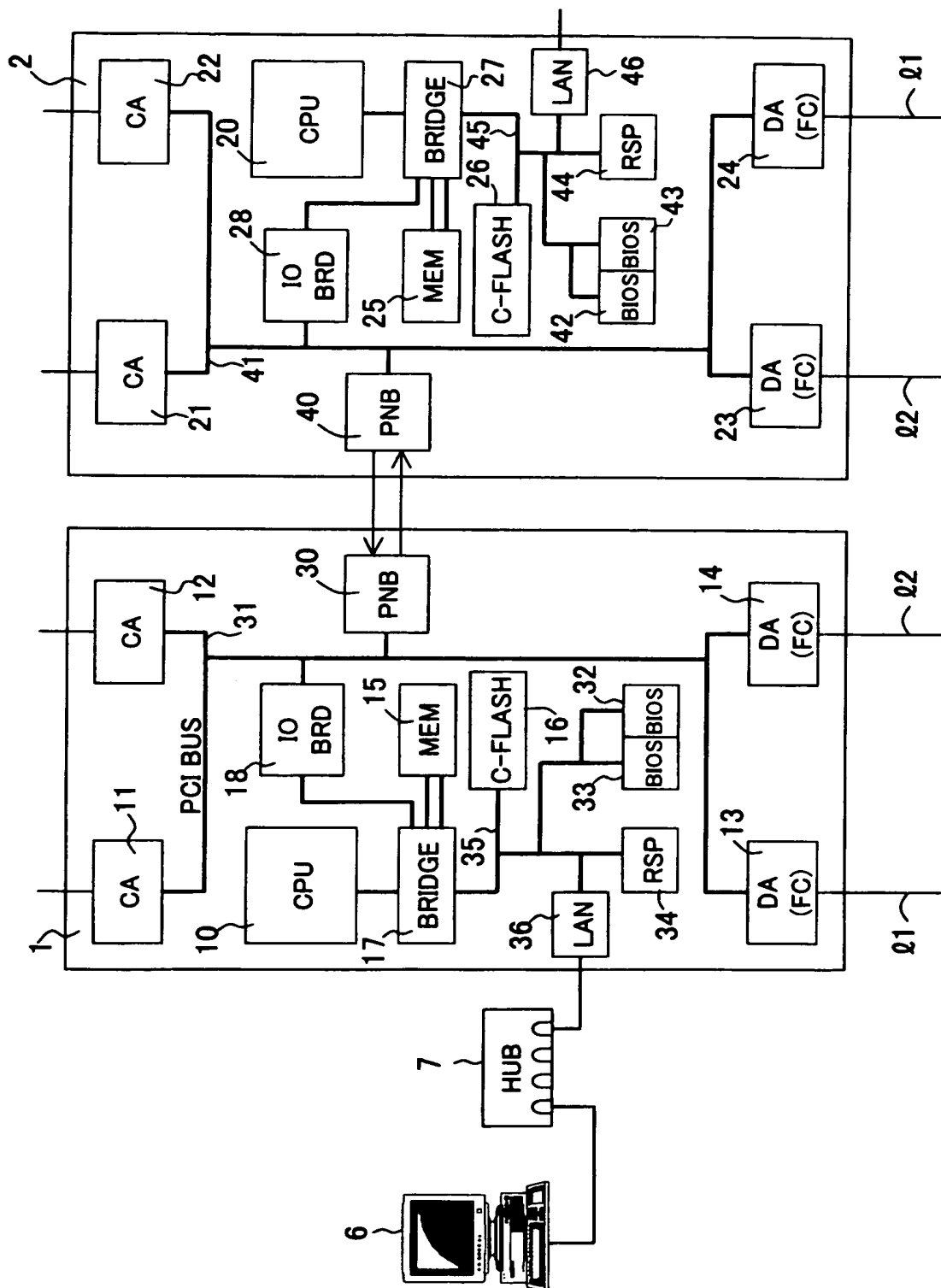
【図 3】



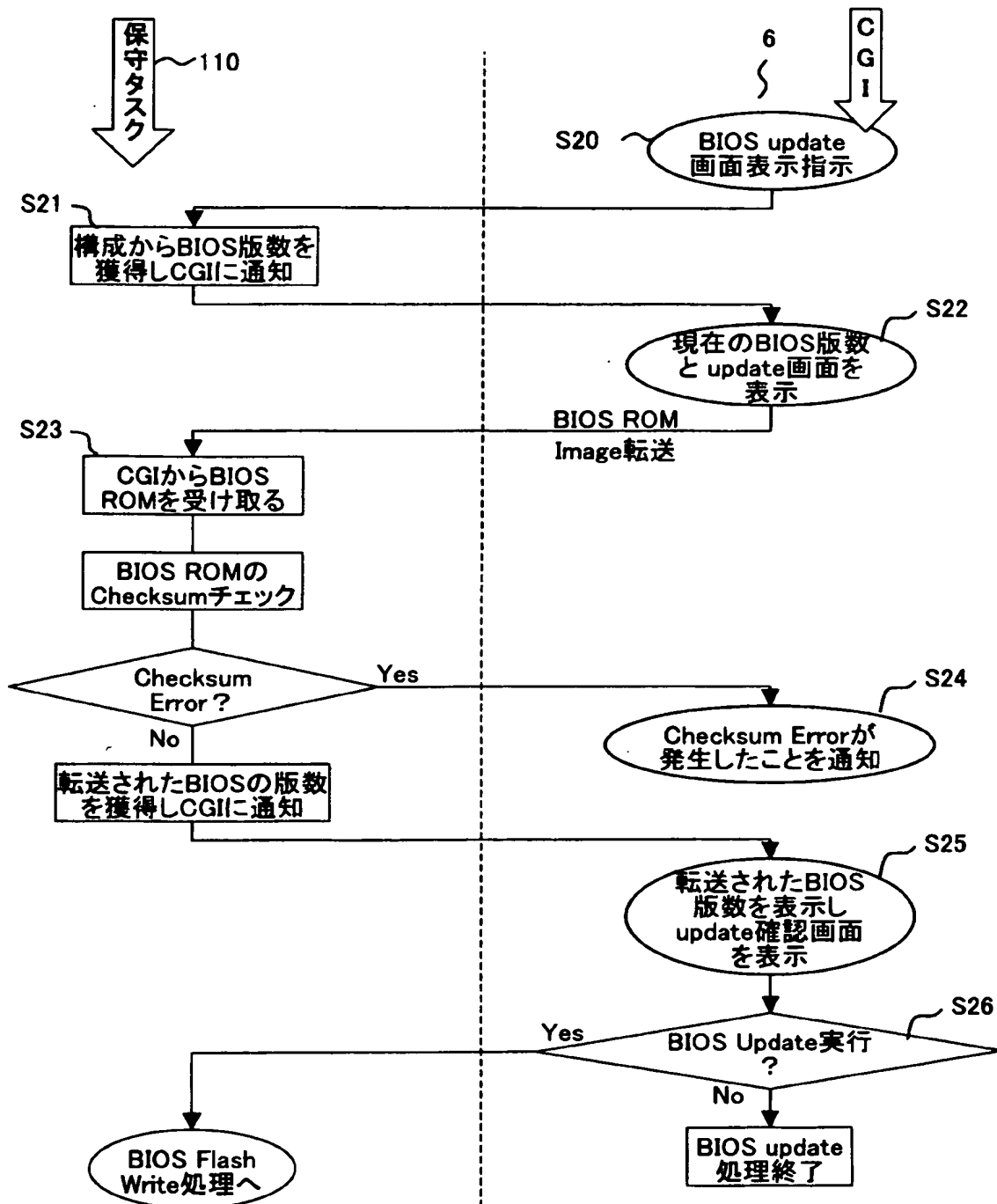
【図 4】



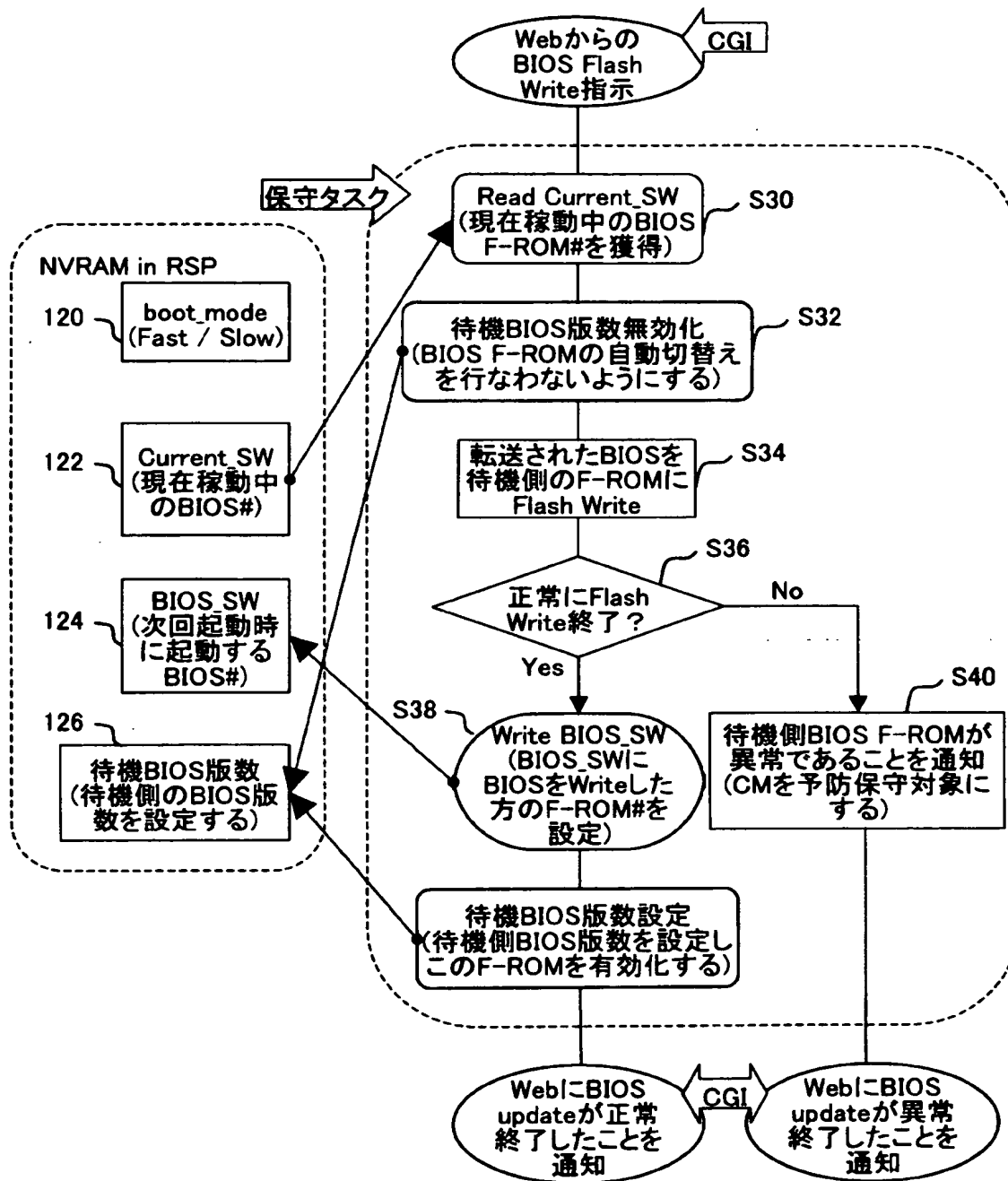
【図 5】



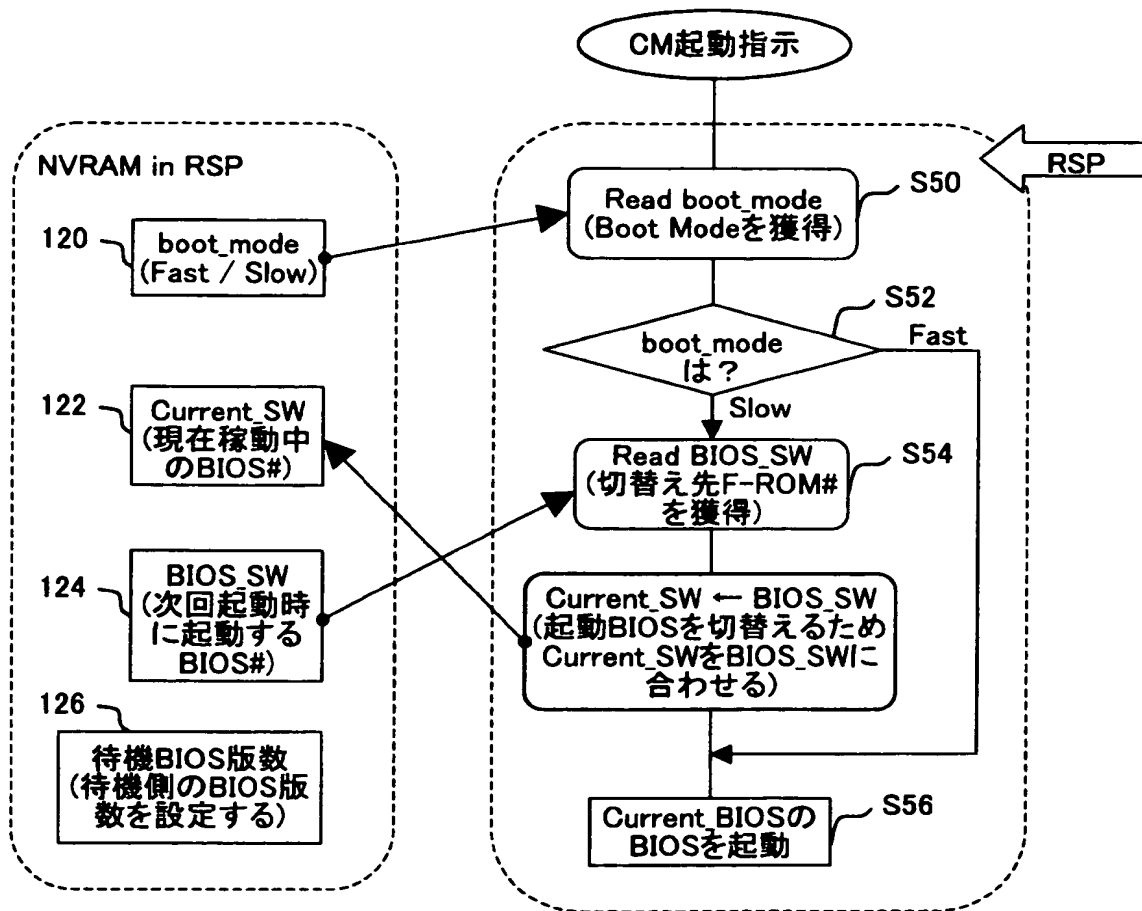
【図 6】



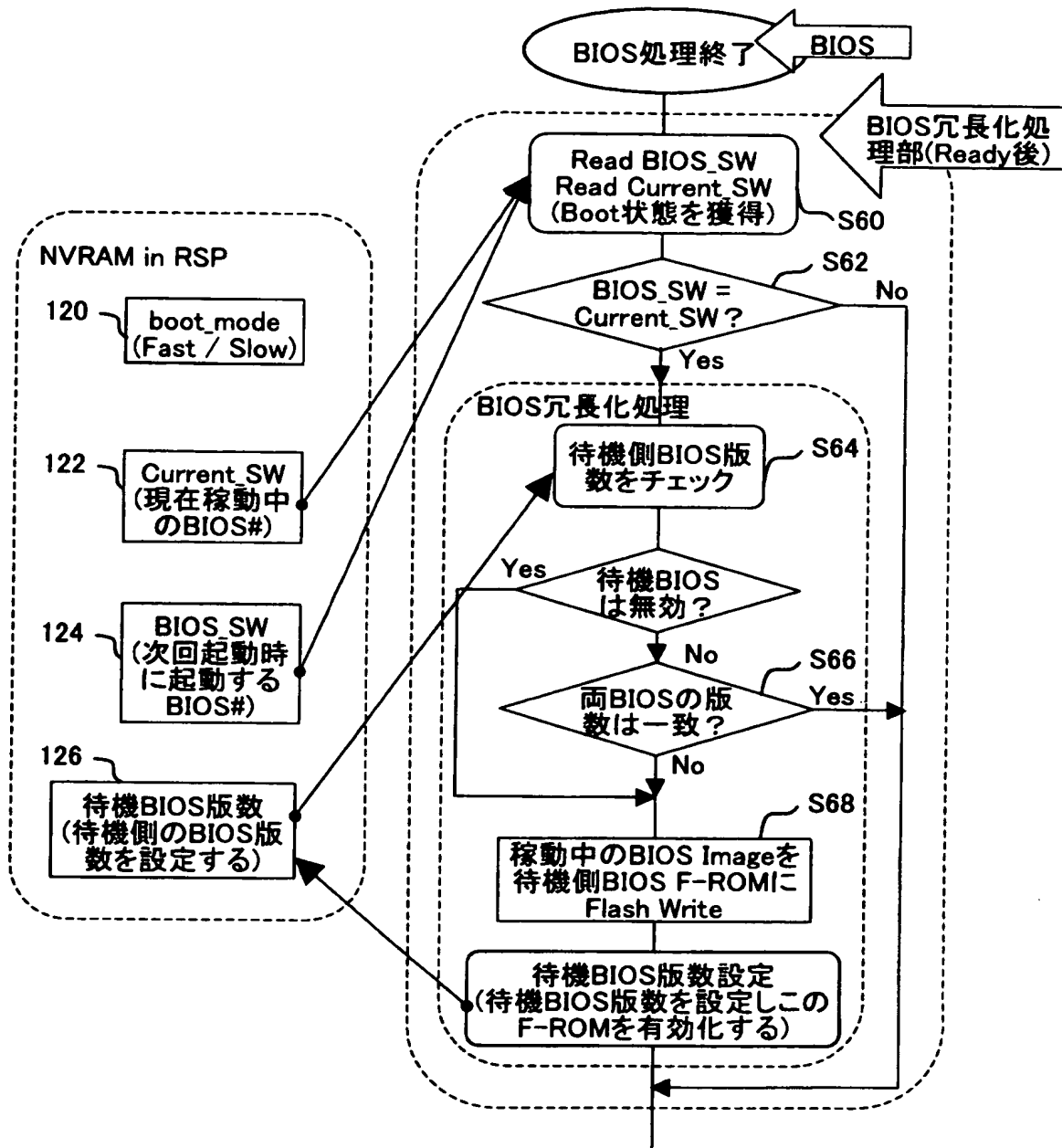
【図 7】



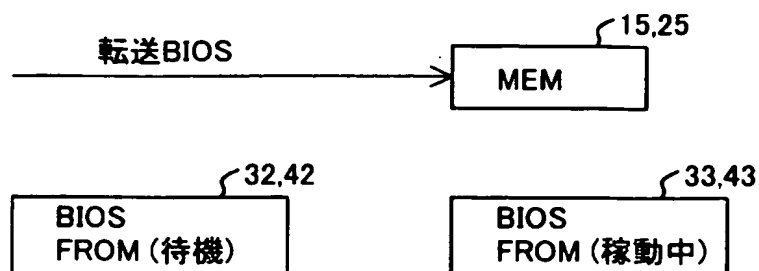
【図 8】



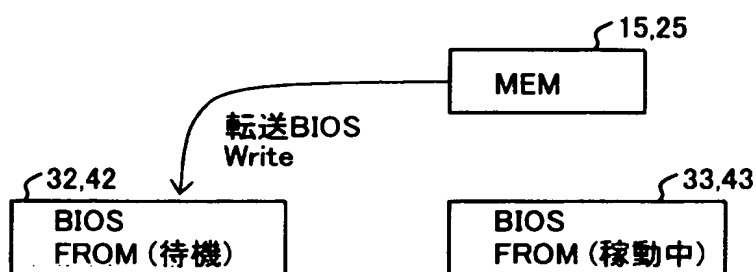
【図 9】



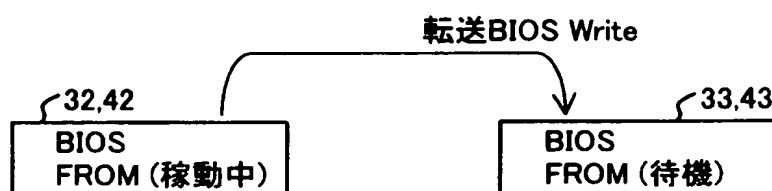
【図 10】



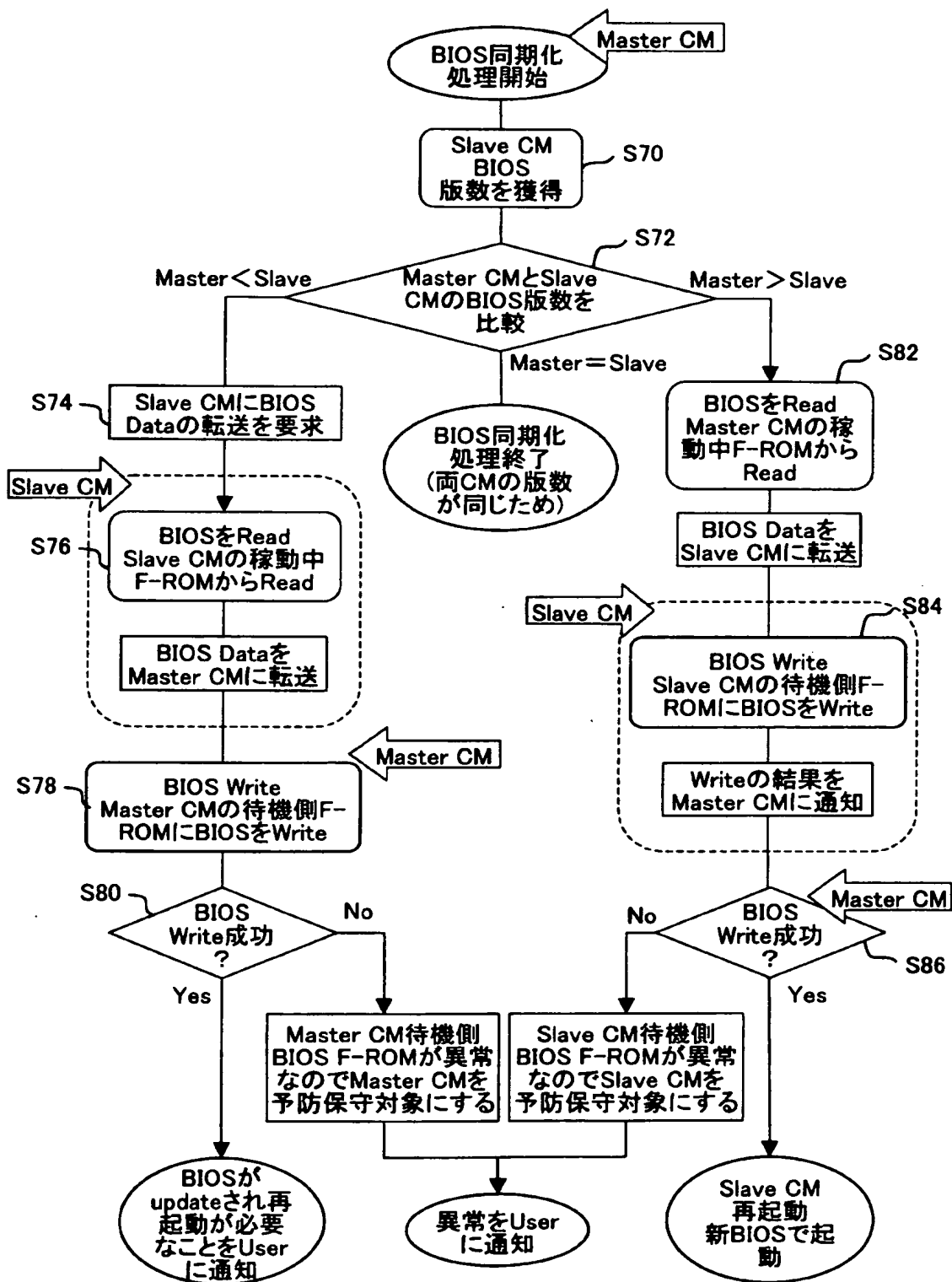
【図 11】



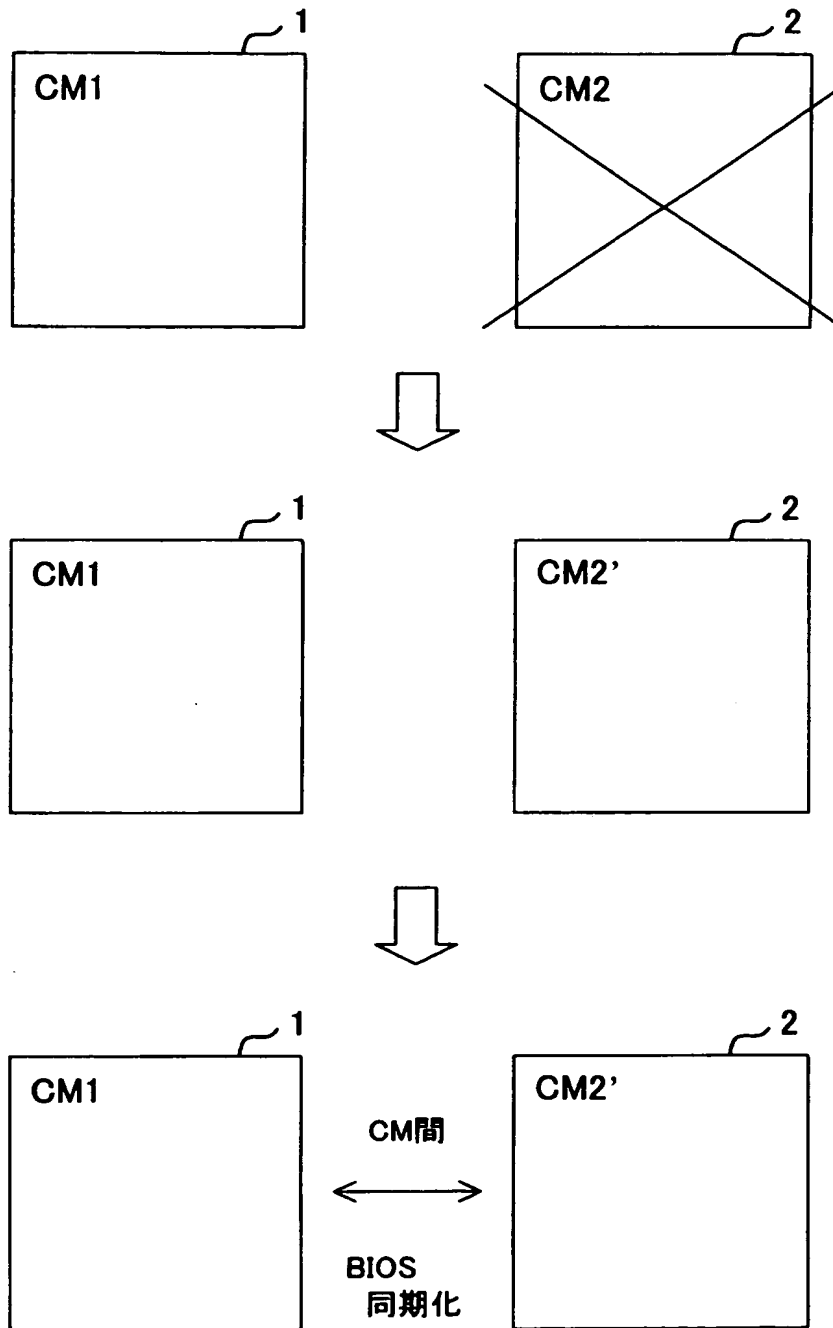
【図 12】



【図13】



【図 14】



【書類名】 要約書

【要約】

【課題】 OSがハードウェアを使用できる環境設定を行うBIOSを冗長管理する方法において、CPUステッピング変更に応じたBIOSアップデートを行っても、システム起動不能を防止する。

【解決手段】 BIOSを一对のメモリ（32，33）で冗長管理し、BIOSのブート不能時に、待機中メモリに切り換え、システム起動不能を防止する。又、CPUステッピング変更に応じたBIOSアップデート時に、同時に2つのBIOSメモリに、ライトせず、待機側のみにライトし、現在稼動中のBIOSを書き換えない。このため、アップデートに失敗した時に、現在稼動中のBIOSで起動でき、システム起動不能になるのを防ぐことができる。

【選択図】 図9

特願 2 0 0 2 - 3 6 5 6 1 8

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 2 2 3]

1. 変更年月日

1 9 9 6 年 3 月 2 6 日

[変更理由]

住所変更

住 所

神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名

富士通株式会社

特願 2 0 0 2 - 3 6 5 6 1 8

出 願 人 履 歴 情 報

識別番号

[0 0 0 1 3 6 1 3 6]

1. 変更年月日

1 9 9 0 年 8 月 3 1 日

[変更理由]

新規登録

住 所

石川県河北郡宇ノ気町字宇野気ヌ 9 8 番地の 2

氏 名

株式会社ピーエフユー

2. 変更年月日

2 0 0 3 年 4 月 7 日

[変更理由]

名称変更

住 所

石川県河北郡宇ノ気町字宇野気ヌ 9 8 番地の 2

氏 名

株式会社 P F U